greatlearning
Power Ahead

# Business Presentation

Presenter : Esteban Ordenes

# Contents

1. Data Definition and perform an Exploratory Data Analysis
2. Illustration of the insights based on EDA
3. Model building - Logistic Regression
4. Model building - Regularization, Oversampling and Undersampling
5. Hyperparameter tuning using grid search
6. Hyperparameter tuning using random search
7. Model performance evaluation - Conclusion
8. Actionable Insights & Recommendations

# Business Problem Overview and Solution Approach

- CreditCard Users Churn Prediction

- Core business idea
  - The Thera bank recently saw a steep decline in the number of users of their credit card, credit cards are a good source of income for banks because of different kinds of fees charged by the banks like annual fees, balance transfer fees, and cash advance fees, late payment fees, foreign transaction fees, and others. Some fees are charged to every user irrespective of usage, while others are charged under specified circumstances.
  - Customers' leaving credit cards services would lead bank to loss, so the bank wants to analyze the data of customers and identify the customers who will leave their credit card services and reason for same – so that bank could improve upon those areas

- Problem to tackle
  - Build a classification model that will help the bank improve their services so that customers do not renounce their credit cards.

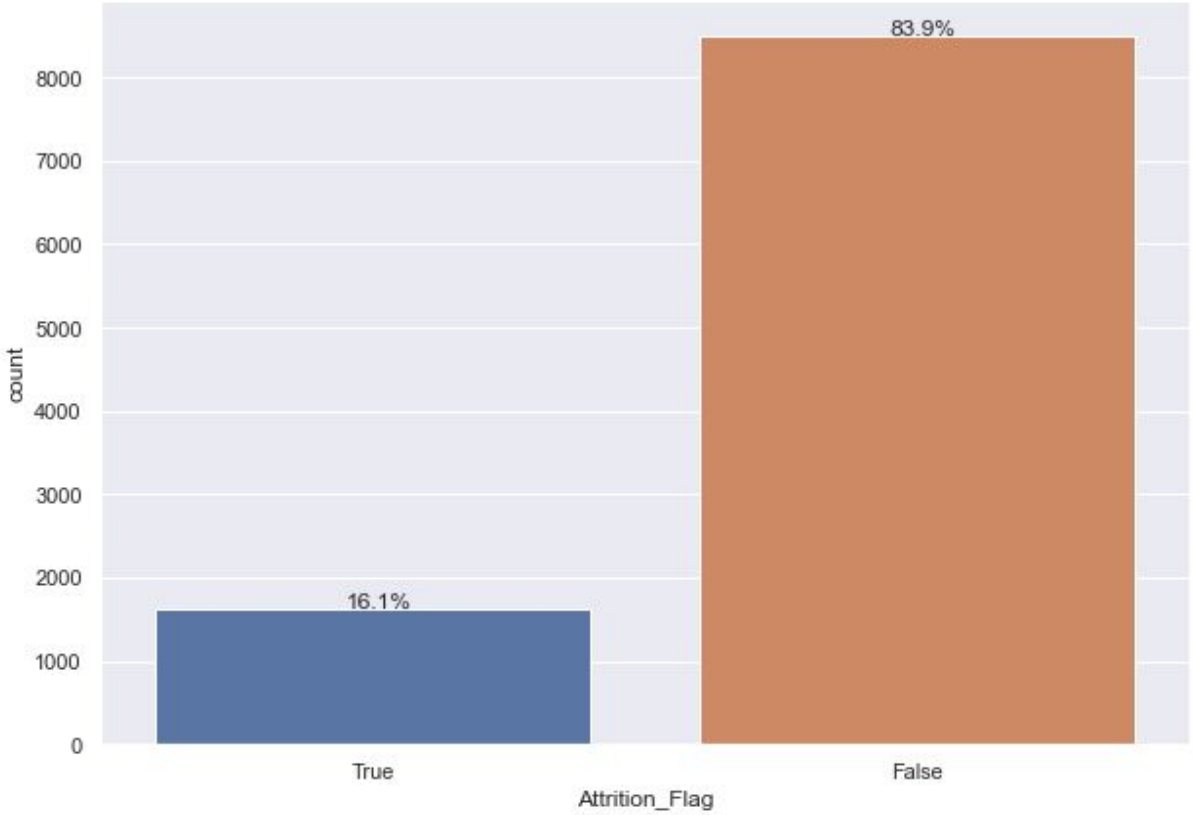
# Data Overview

- **Data Dictionary**
  - CLIENTNUM: Client number. Unique identifier for the customer holding the account
  - Attrition_Flag: Internal event (customer activity) variable - if the account is closed then 1 else 0
  - Customer_Age: Age in Years
  - Gender: Gender of the account holder
  - Dependent_count: Number of dependents
  - Education_Level: Educational Qualification of the account holder
  - Marital_Status: Marital Status of the account holder
  - Income_Category: Annual Income Category of the account holder
  - Card_Category: Type of Card

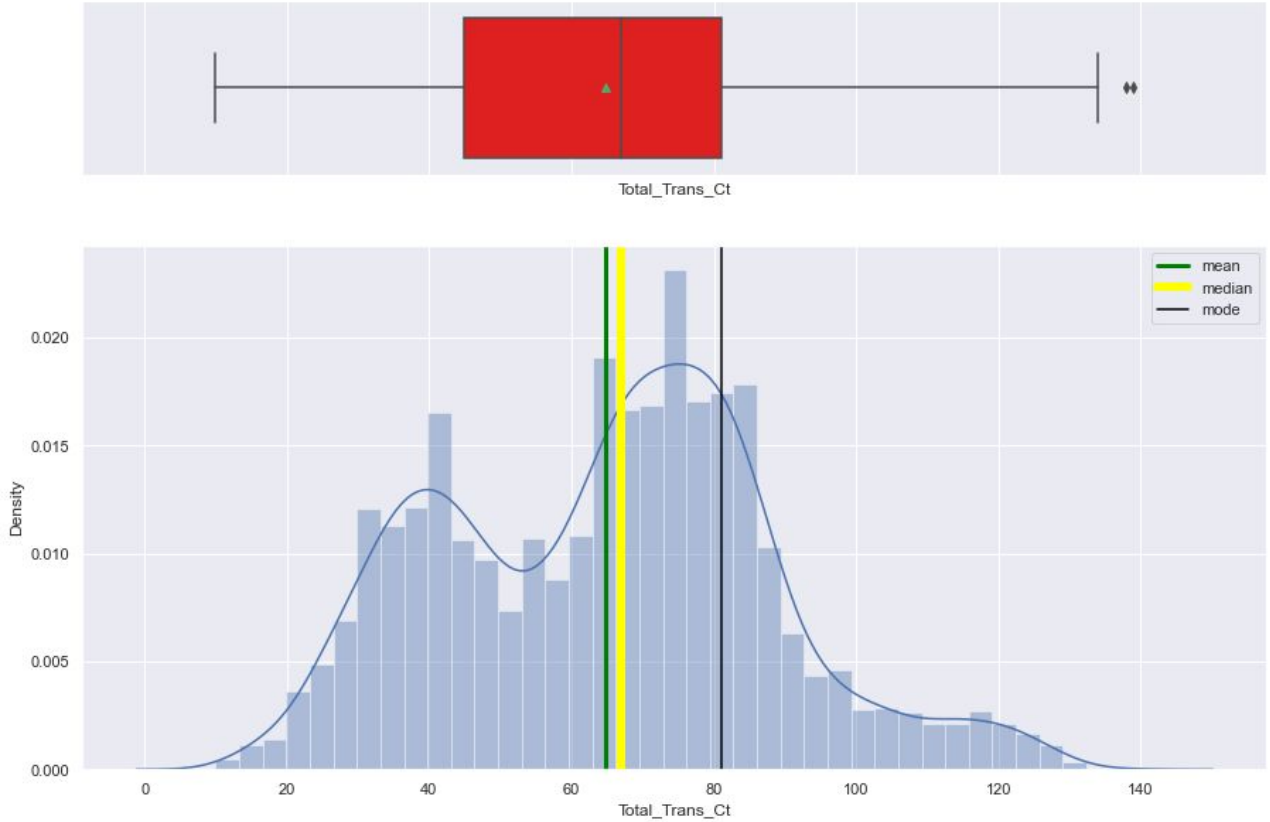
# Data Overview

- **Data Dictionary**
  - Months_on_book: Period of relationship with the bank
  - Total_Relationship_Count: Total no. of products held by the customer
  - Months_Inactive_12_mon: No. of months inactive in the last 12 months
  - Contacts_Count_12_mon: No. of Contacts in the last 12 months
  - Credit_Limit: Credit Limit on the Credit Card
  - Total_Revolving_Bal: The balance that carries over from one month to the next is the revolving balance
  - Avg_Open_To_Buy: Open to Buy refers to the amount left on the credit card to use (Average of last 12 months)
  - Total_Trans_Amt: Total Transaction Amount (Last 12 months)
  - Total_Trans_Ct: Total Transaction Count (Last 12 months)
  - Total_Ct_Chng_Q4_Q1: Ratio of the total transaction count in 4th quarter and the total transaction count in 1st quarter
  - Total_Amt_Chng_Q4_Q1: Ratio of the total transaction amount in 4th quarter and the total transaction amount in 1st quarter
  - Avg_Utilization_Ratio: Represents how much of the available credit the customer spent

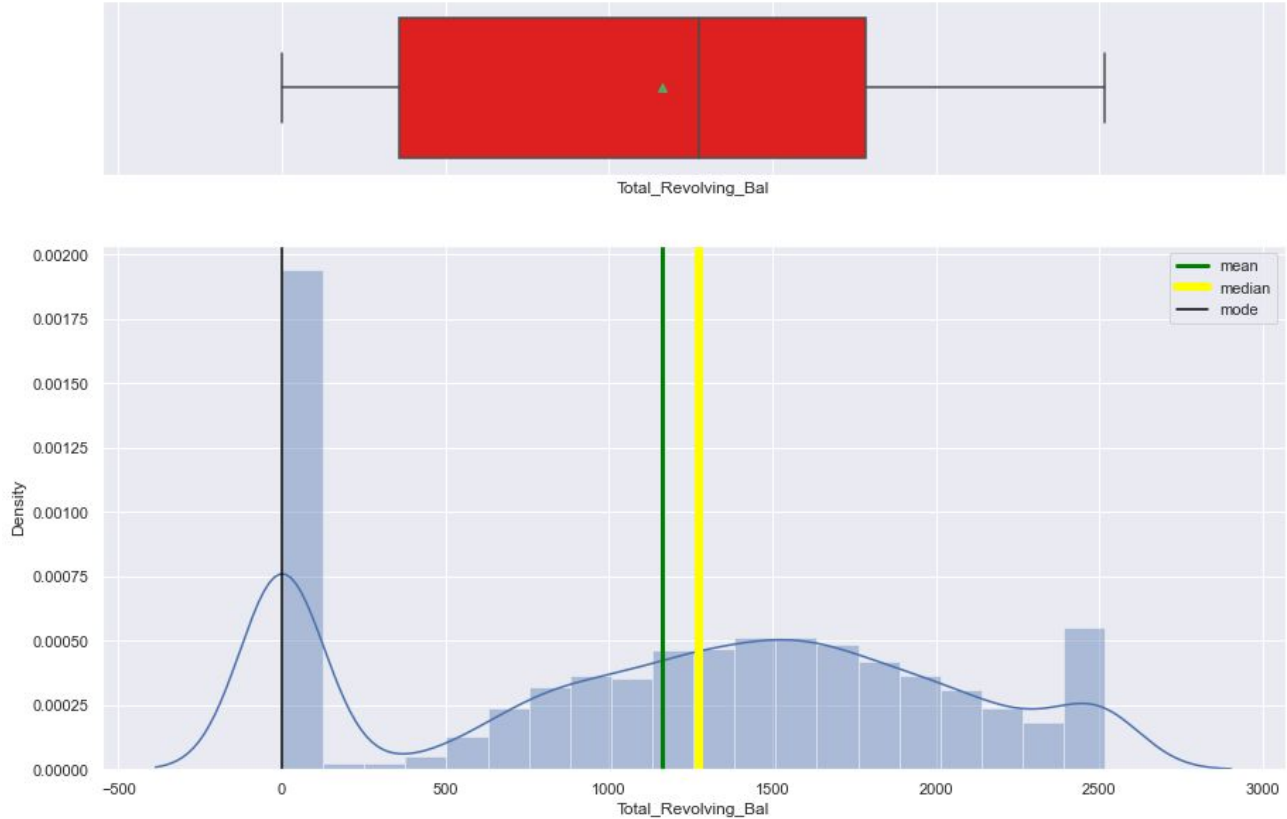
# EDA - Target Variable : Attrition_Flag



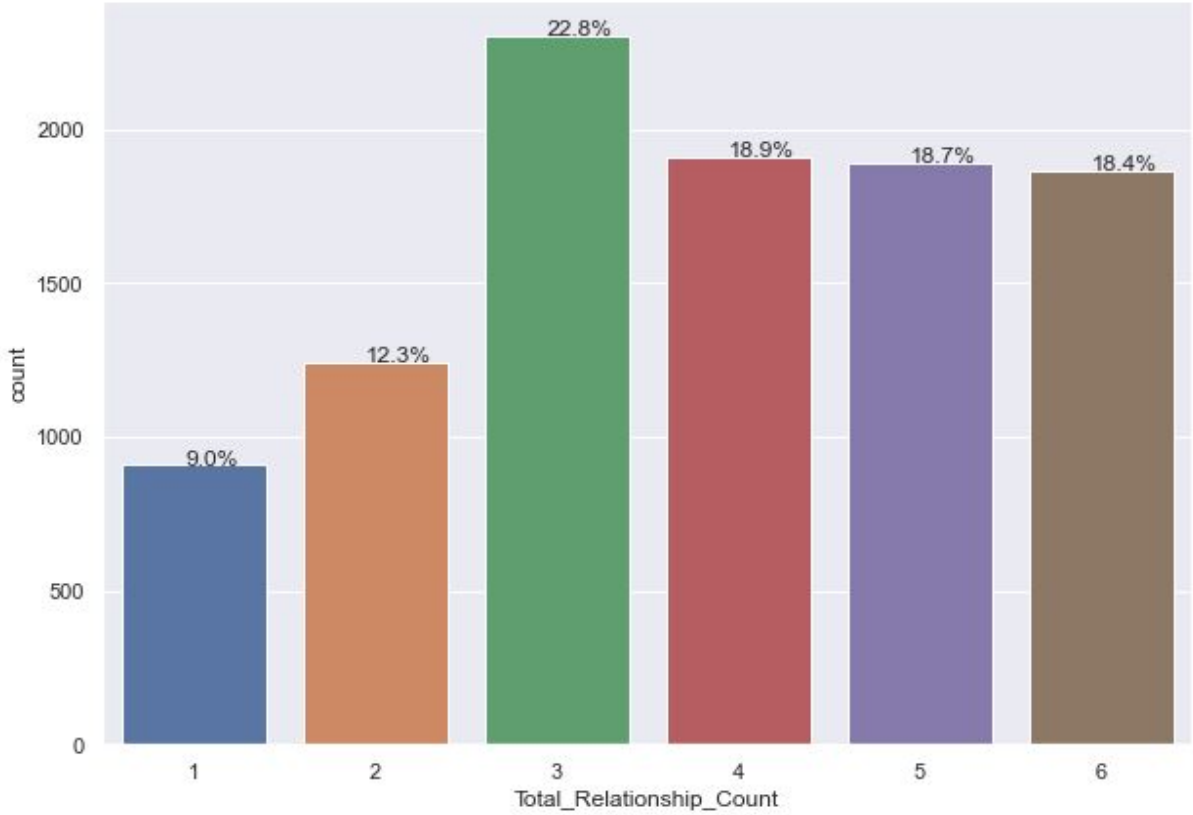
# EDA - Feature : Total_Trans_Ct



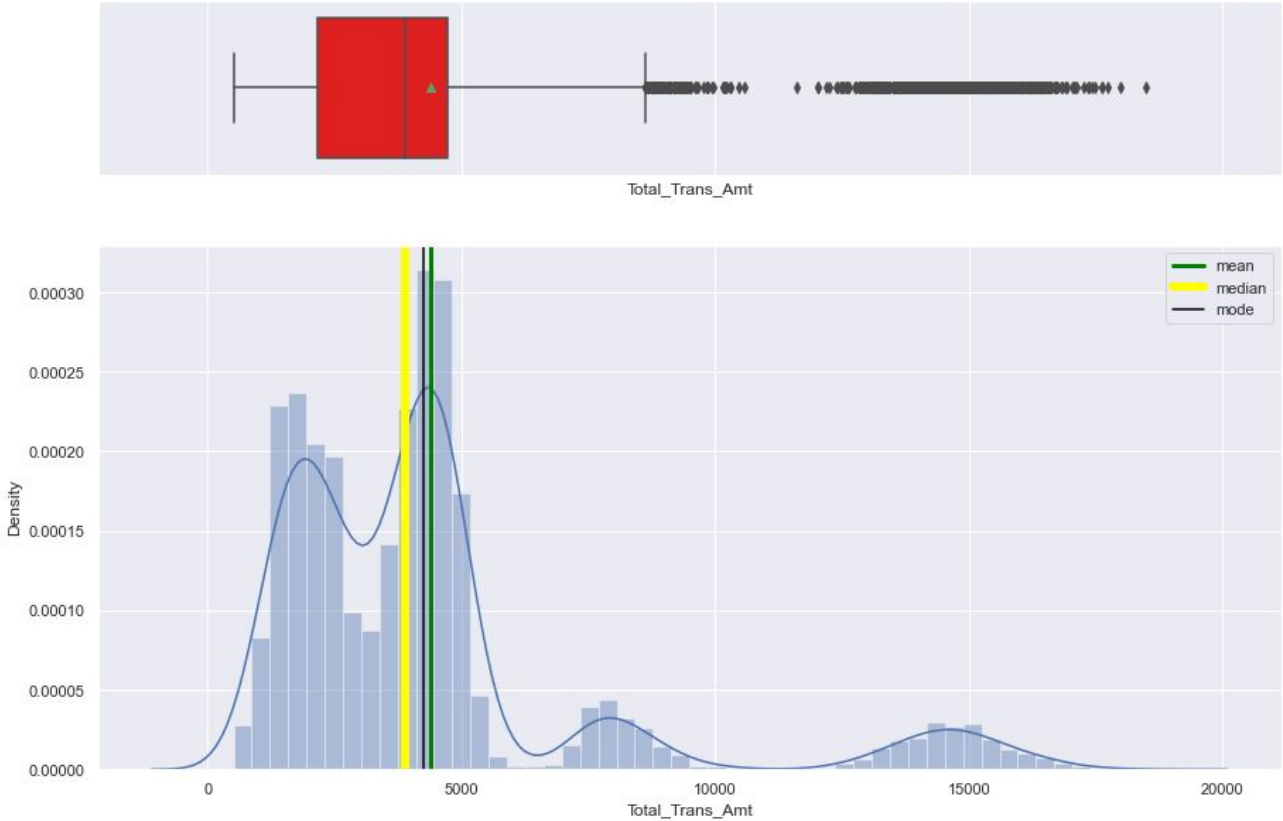
# EDA - Feature : Total_Revolving_Bal



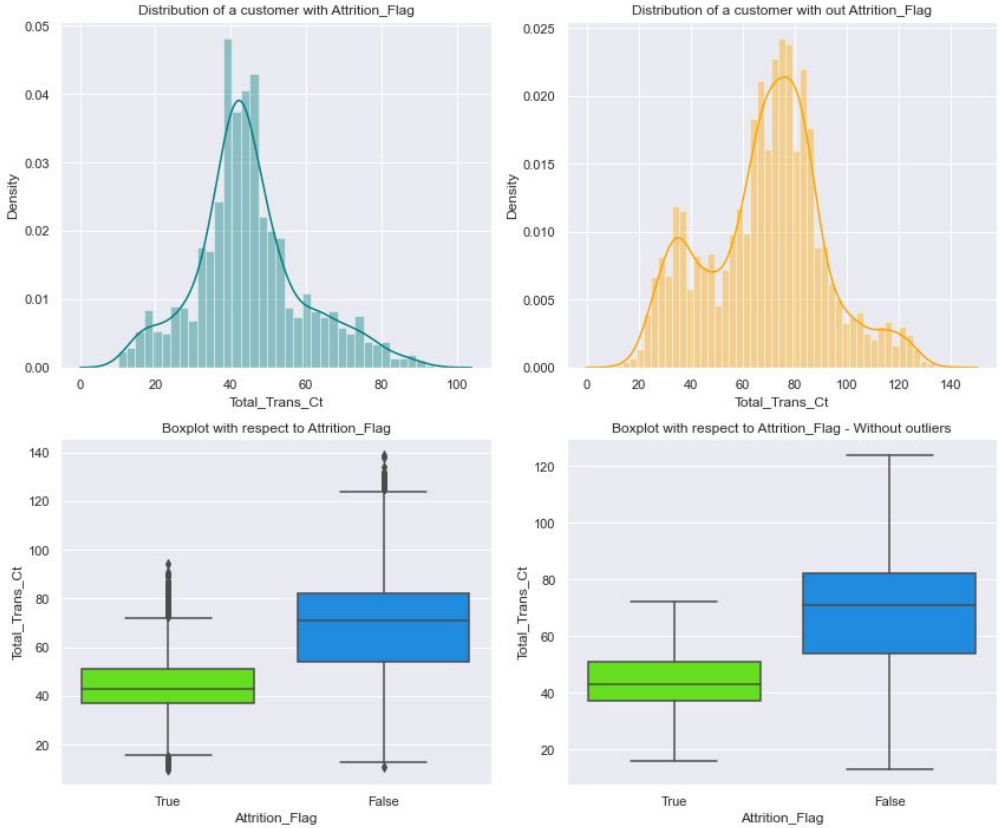
# EDA - Feature : Total_Relationship_Count



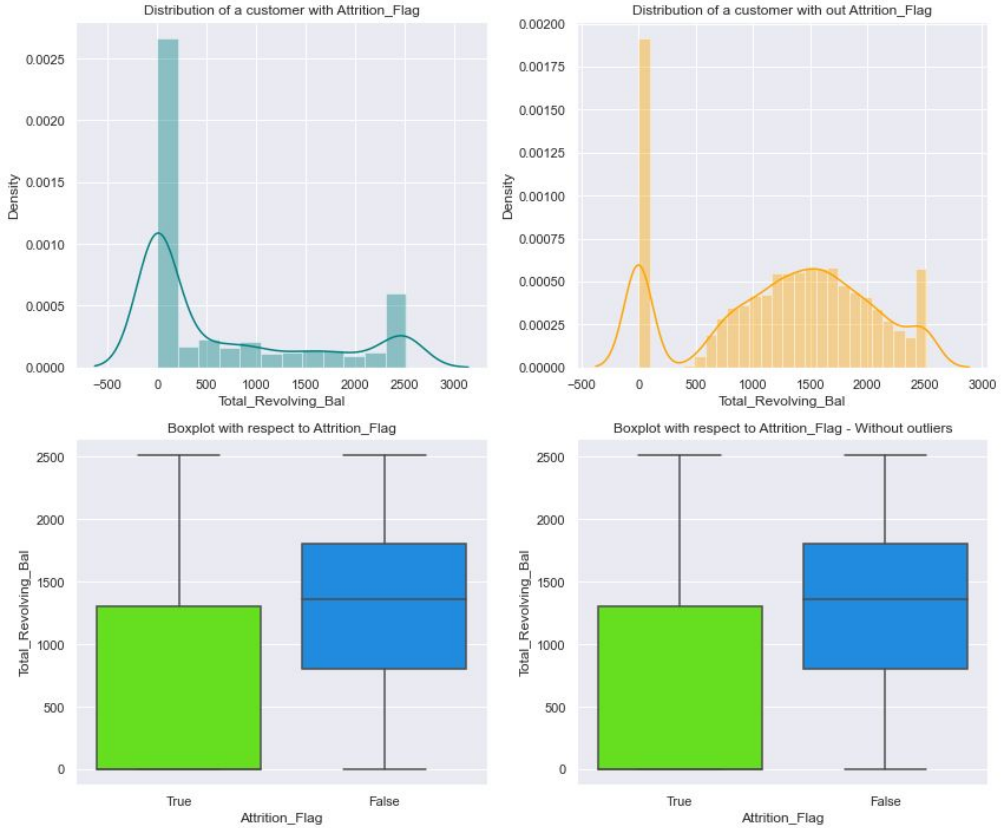
# EDA - Feature : Total_Trans_Amount



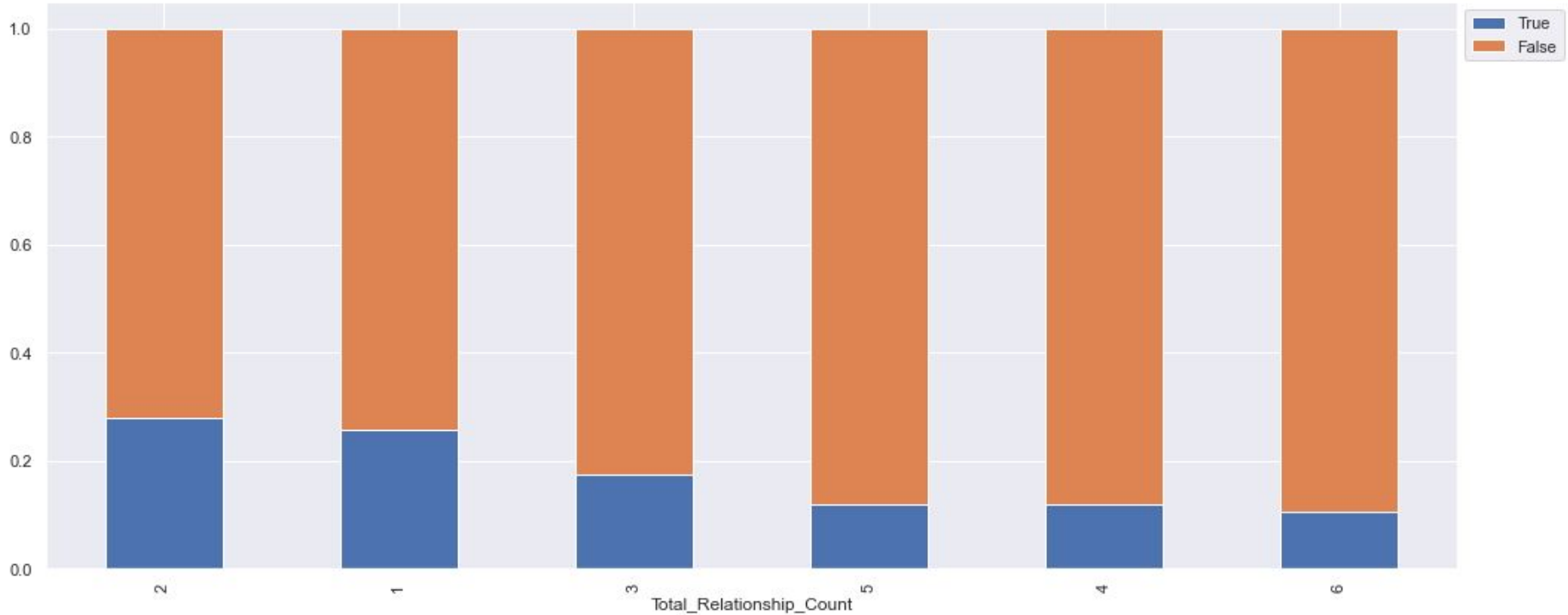
# EDA - Total_Trans_Ct vs Attrition_Flag



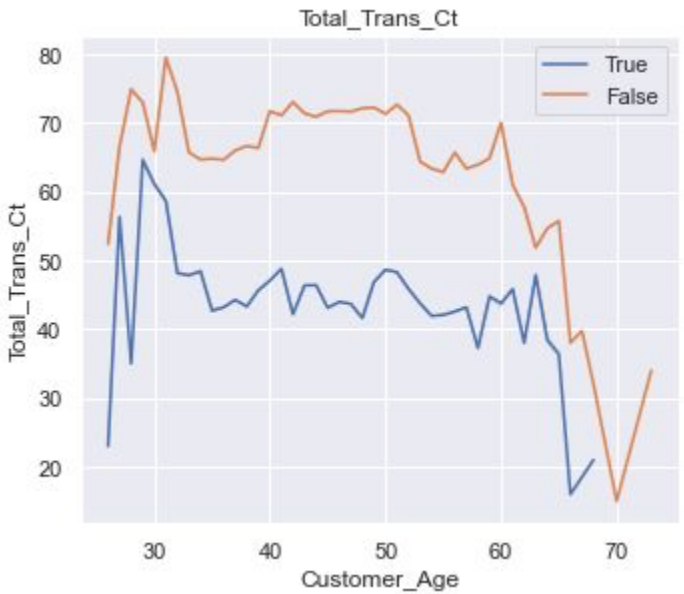
# EDA - **Total_Revolving_Bal vs Attrition_Flag**

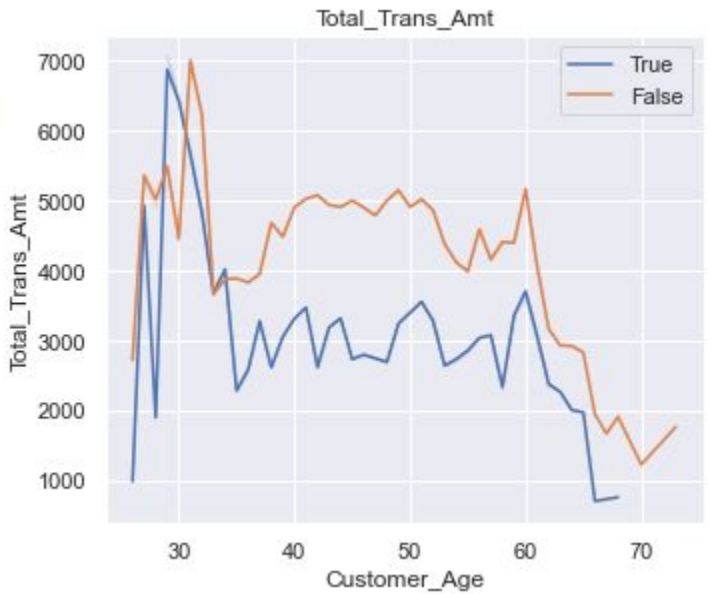


# EDA - Total_Relationship_Count vs Attrition_Flag

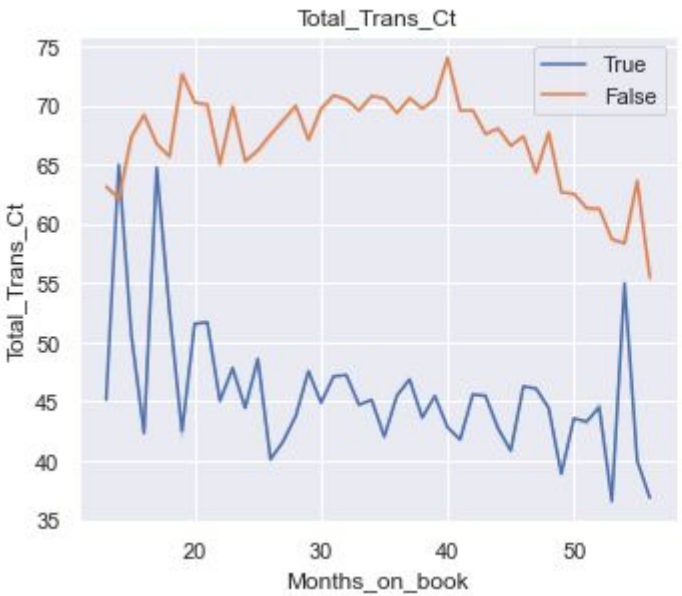


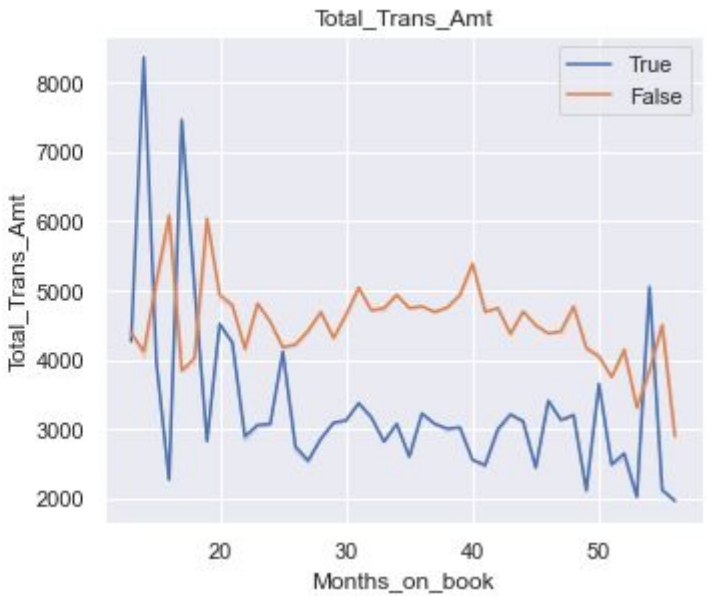
# EDA - Total_Trans_Ct/Total_Trans_Amt vs Customer_Age and Attrition_Flag



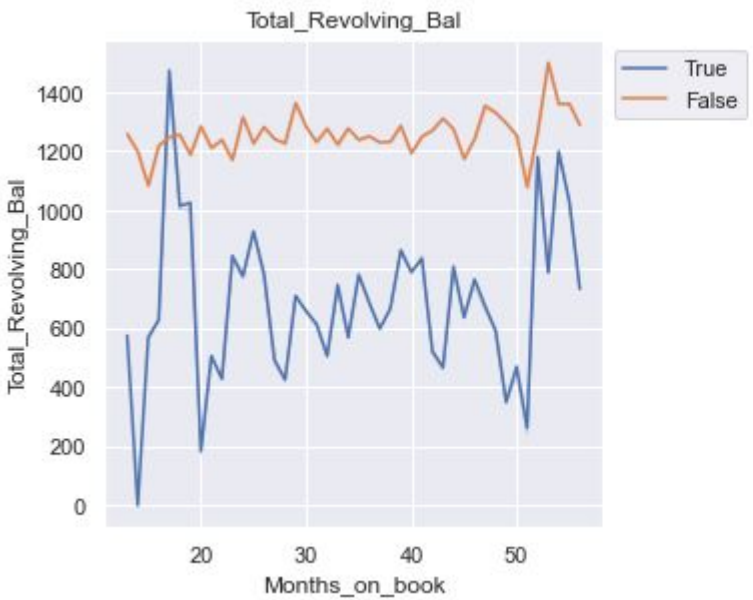



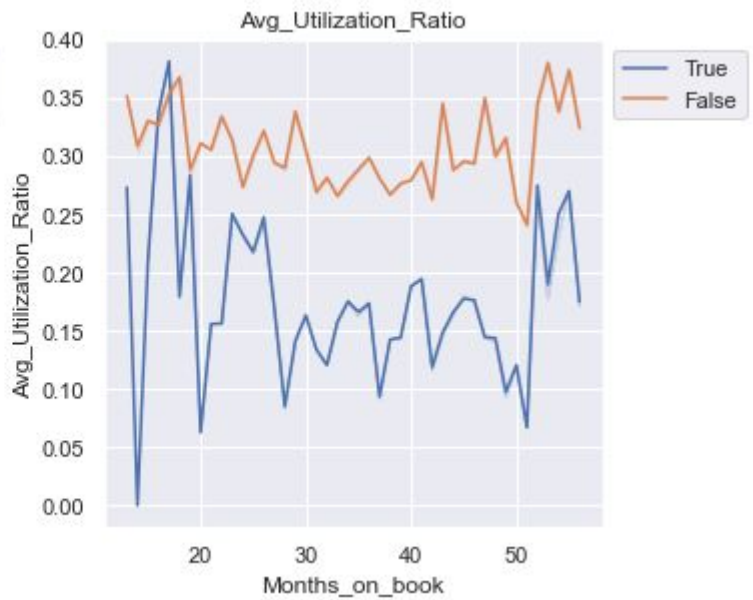
# EDA - Total_Trans_Ct/Total_Trans_Amt vs Months_on_book and Attrition_Flag



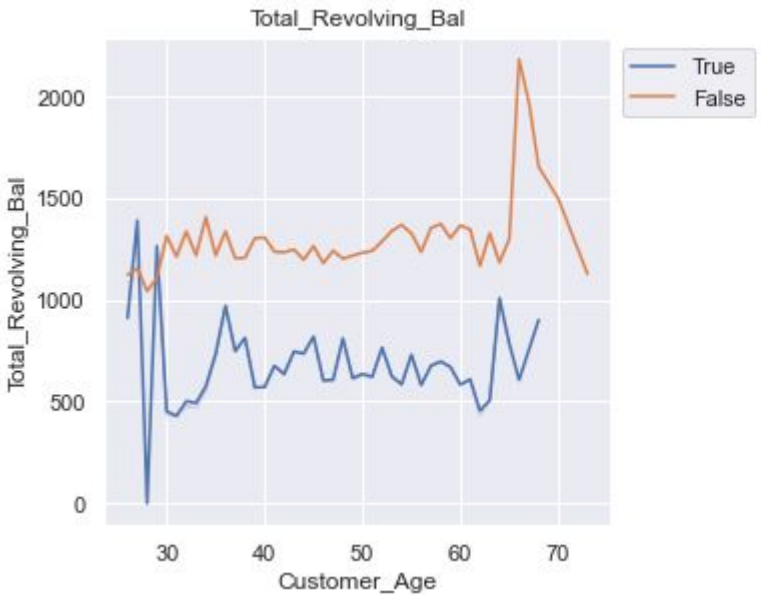



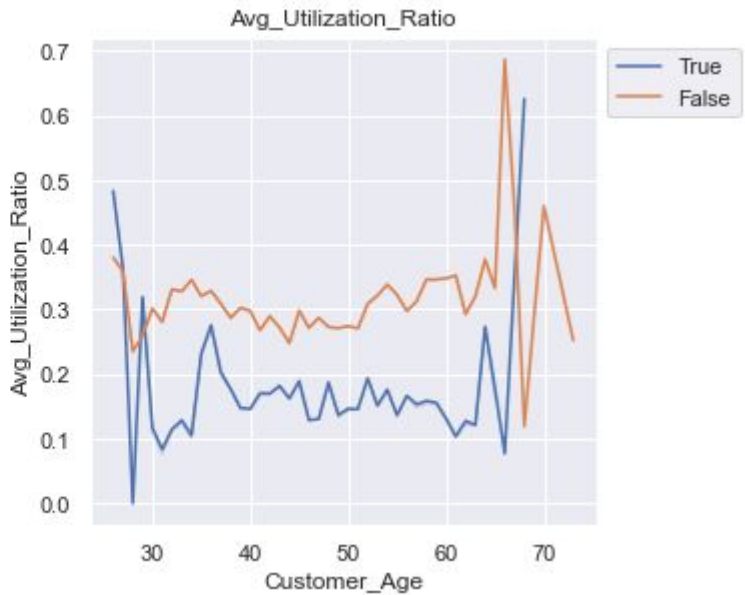
# EDA - Total_Revolving_Ba/'Avg_Utilization_Ratio vs Months_on_book and Attrition_Flag



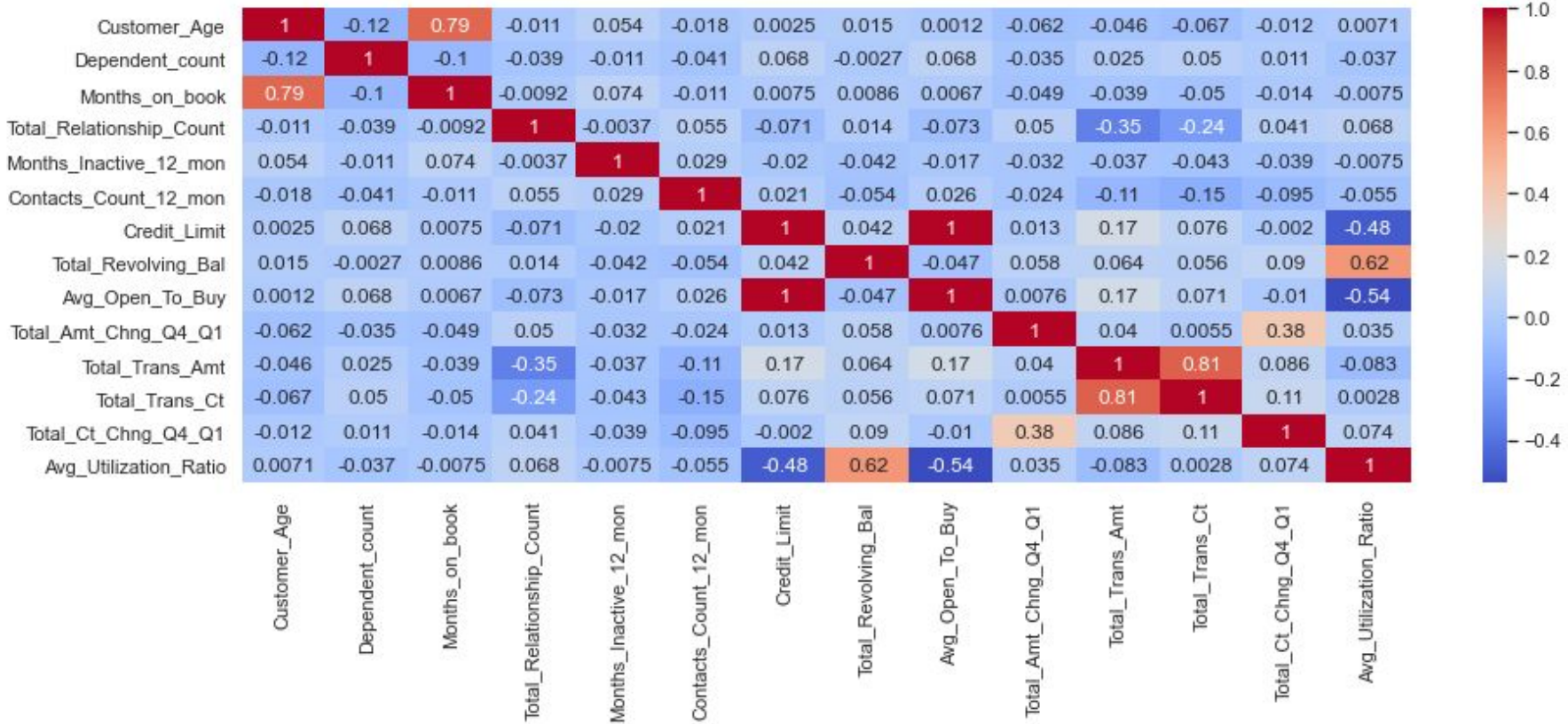



# EDA - Total_Revolving_Ba/'Avg_Utilization_Ratio vs Customer_Age and Attrition_Flag

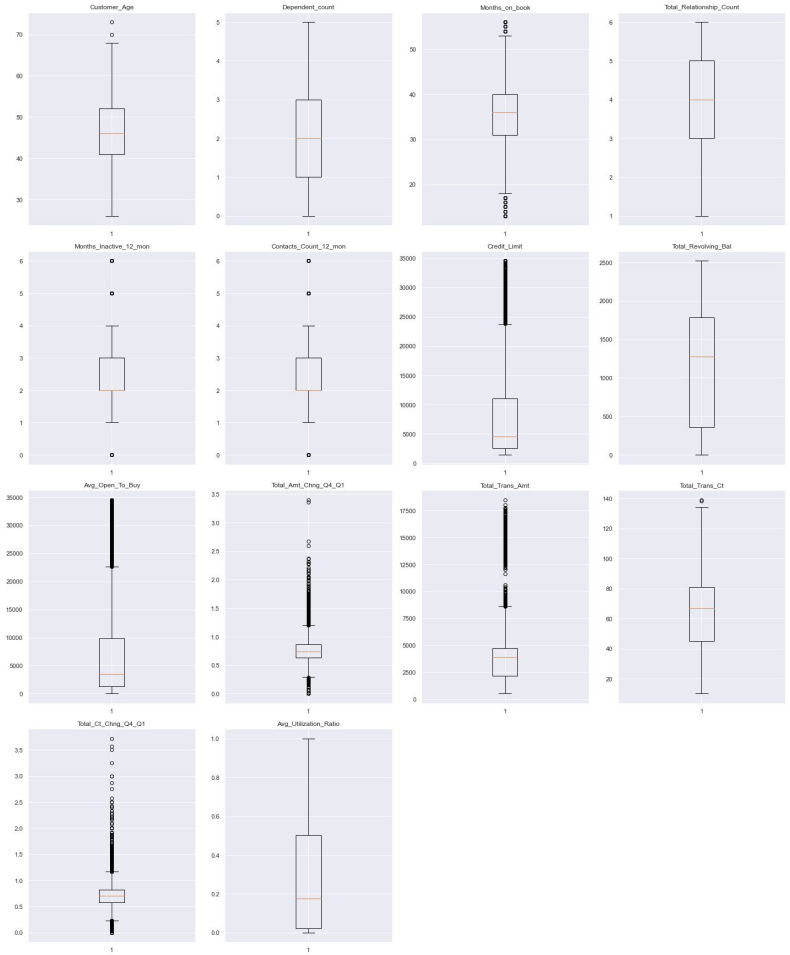


# EDA



# EDA - Outlier Analysis



# Model building - Logistic Regression

Accuracy on training set : 0.8844525959367946
Accuracy on test set : 0.8907535373478118
Recall on training set : 0.47058823529411764
Recall on test set : 0.4918032786885246
Precision on training set : 0.7127659574468085
Precision on test set : 0.7407407407407407

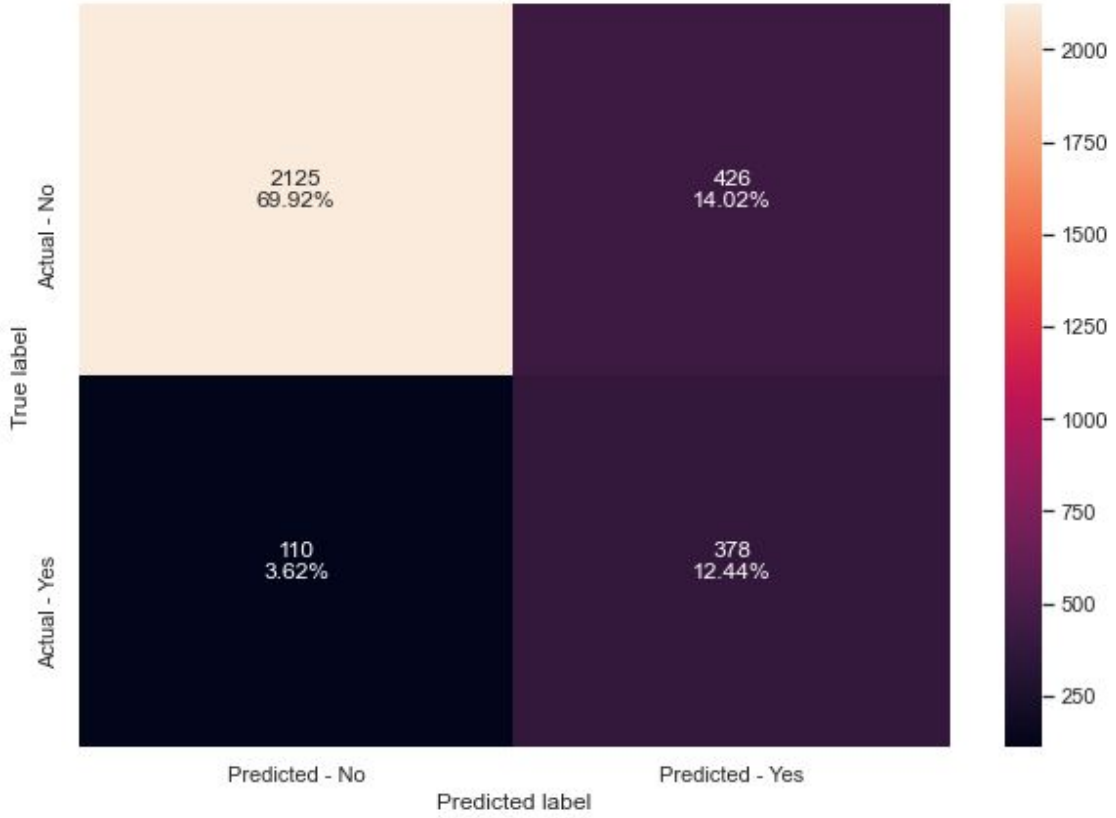
* Logistic Regression has given a generalized performance on training and test set.

* Recall is very low, we can try oversampling (increase training data) to see if the model performance can be improved.

# Model building - Logistic Regression on oversampled data

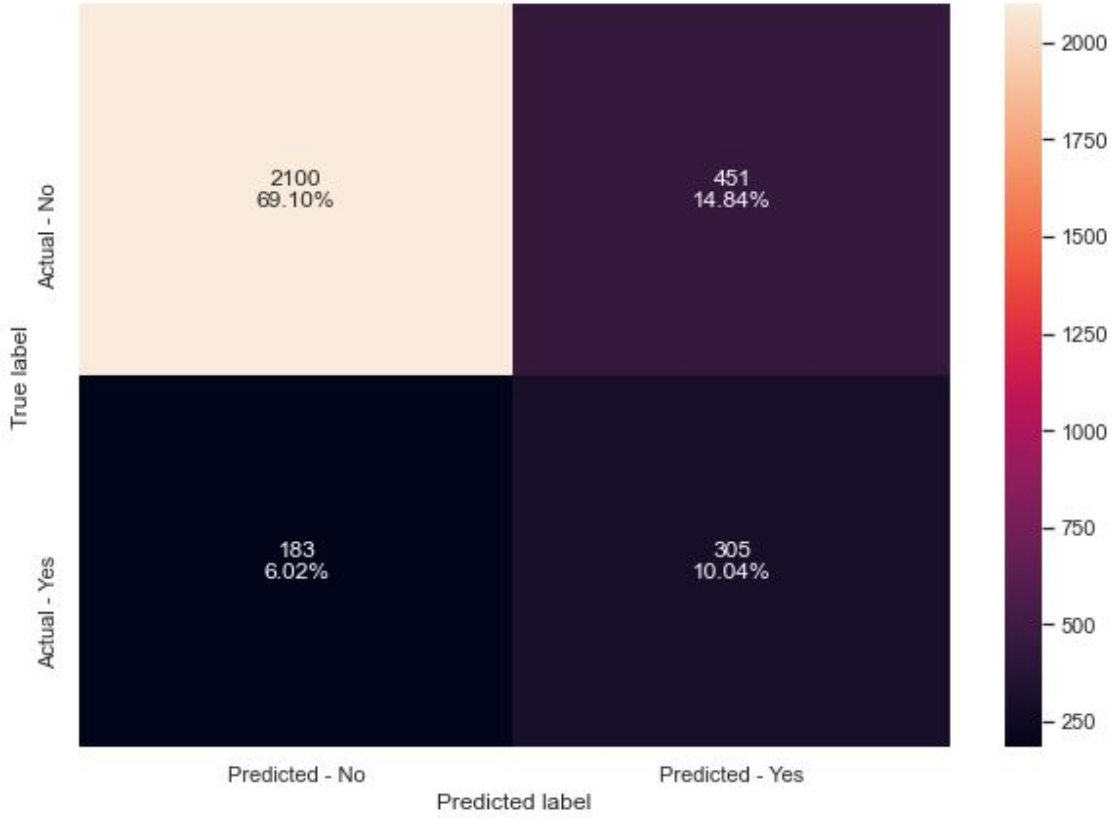
Accuracy on training set : 0.83761976802824
Accuracy on test set : 0.8236261928265877
Recall on training set : 0.8411497730711044
Recall on test set : 0.7745901639344263
Precision on training set : 0.8352528793189785
Precision on test set : 0.4701492537313433

- Performance on the training set improved but the model is not able to replicate the same for the test set.
- Model is overfitting.



# Model building - Logistic Regression with Regularization

Accuracy on training set : 0.7254160363086233
Accuracy on test set : 0.7913787430075683
Recall on training set : 0.6396032946713733
Recall on test set : 0.625
Precision on training set : 0.7721185064935064
Precision on test set : 0.40343915343915343




- After regularization, overfitting has reduced to some extent and the model is also performing well.

# Model building - Logistic Regression on undersampled data

Accuracy on training set : 0.810359964881475
Accuracy on test set : 0.8147416913458374
Recall on training set : 0.8121158911325724
Recall on test set : 0.8114754098360656
Precision on training set : 0.8092738407699037
Precision on test set : 0.45674740484429066

- Model has given a generalized performance on training and test set.
- Model performance has improved using downsampling - Logistic regression is now able to differentiate well between positive and negative classes

# Logistic Regression Model Performance Summary

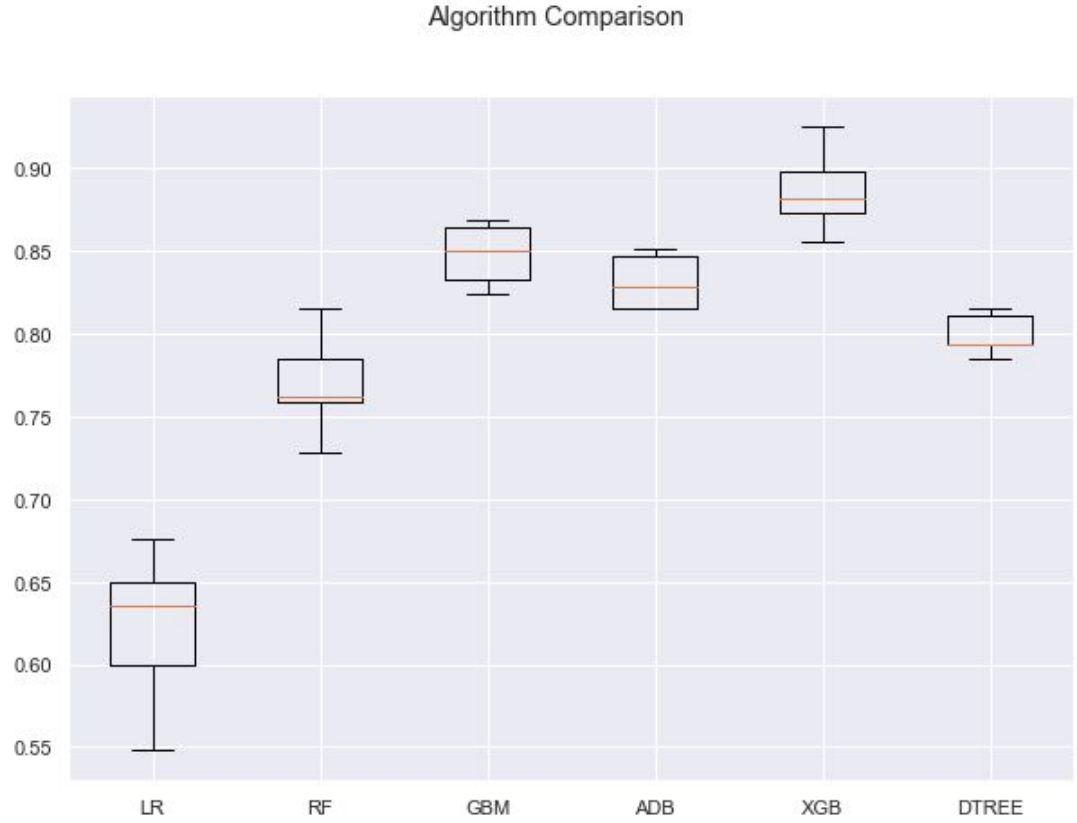
| | Model | Train_Accuracy | Test_Accuracy | Train_Recall | Test_Recall | Train_Precision | Test_Precision |
|---|---|---|---|---|---|---|---|
| **0** | Logistic Regression | 0.884453 | 0.890754 | 0.470588 | 0.491803 | 0.712766 | 0.740741 |
| **1** | Logistic Regression on Oversampled data | 0.837620 | 0.823626 | 0.841150 | 0.774590 | 0.835253 | 0.470149 |
| **2** | Logistic Regression-Regularized (Oversampled d... | 0.725416 | 0.791379 | 0.639603 | 0.625000 | 0.772119 | 0.403439 |
| **3** | Logistic Regression on Undersampled data | 0.810360 | 0.814742 | 0.812116 | 0.811475 | 0.809274 | 0.456747 |

- `Logistic Regression on Undersampled data` has given a generalized performance with the highest recall on test data (0.811475 ).

# Building different models using KFold and cross_val_score with pipelines

LR: 62.15781745111679
RF: 76.99667671381096
GBM: 84.81142283020326
ADB: 83.14282402040342
XGB: 88.67532266790323
DTREE: 79.98376999768142

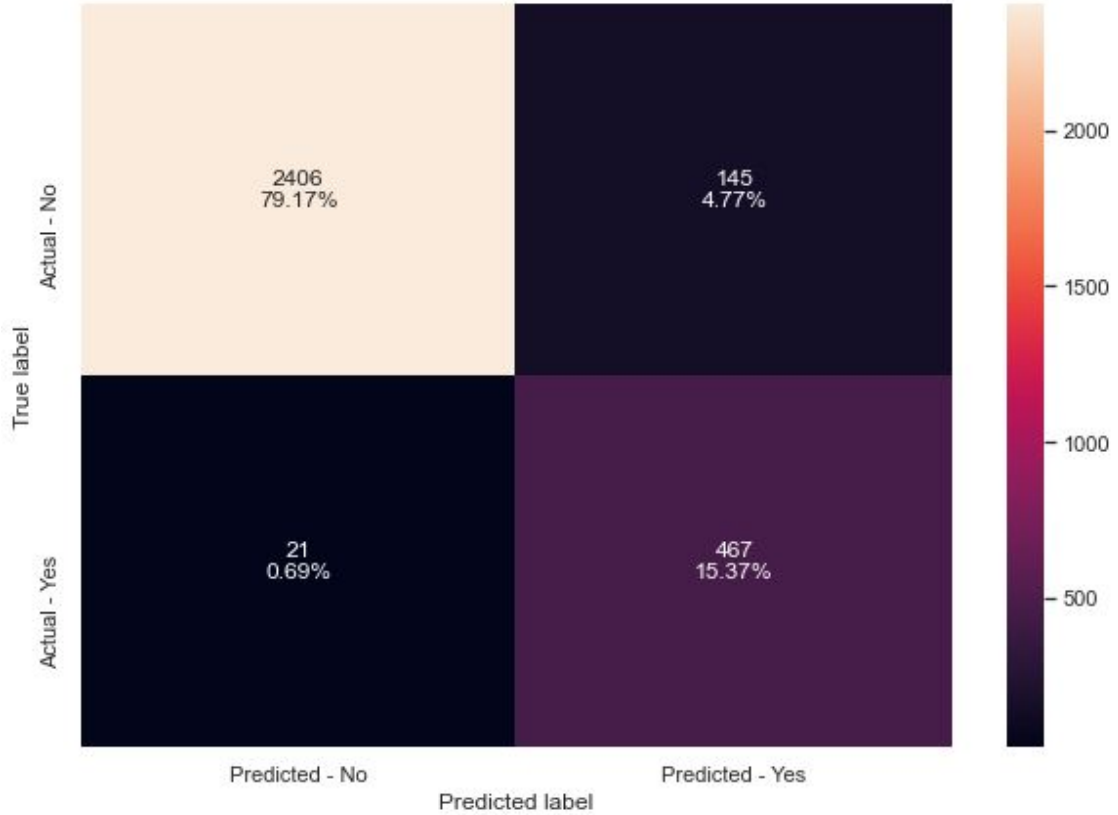
- We can see that XGBoost is giving the highest cross-validated recall followed by GradientBoost and AdaBoost
- The boxplot shows that the performance of all the models is consistent with no outlier.
- We will tune 3 models - XGBoost , GradientBoost , AdaBoost and see if the performance improves.



# Hyperparameter tuning - XGBoost with GridSearchCV

Accuracy on training set : 0.9672686230248307
Accuracy on test set : 0.9453767686739059
Recall on training set : 0.9991220368744512
Recall on test set : 0.9569672131147541
Precision on training set : 0.831263696128561
Precision on test set : 0.7630718954248366

- The test recall has increased by ~10% as compared to the result from cross-validation with default parameters.
- The model is overfitting the training data.

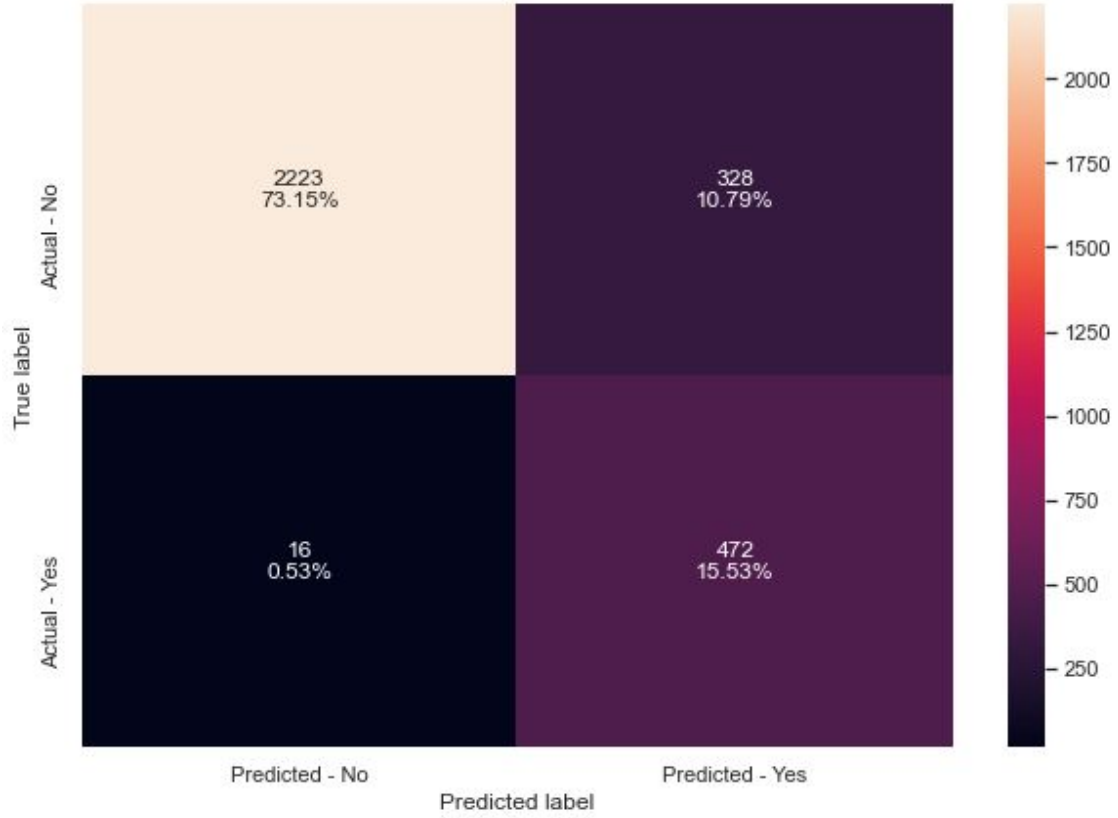


# Hyperparameter tuning - XGBoost with RandomizedSearchCV

Best parameters are
{'xgbclassifier__subsample': 0.8,
'xgbclassifier__scale_pos_weight': 10,
'xgbclassifier__n_estimators': 40,
'xgbclassifier__learning_rate': 0.2,
'xgbclassifier__gamma': 3} with CV
score=0.9420588917226989

Accuracy on training set : 0.897714446952596
Accuracy on test set : 0.8868048700230339
Recall on training set : 0.9859525899912204
Recall on test set : 0.9672131147540983
Precision on training set : 0.6129912663755459
Precision on test set : 0.59

- Random search is giving better results than Grid search.
- The test recall has increased as compared to the test recall from grid search but the accuracy and precision have decreased.



# Hyperparameter tuning - GradientBoost with GridSearchCV

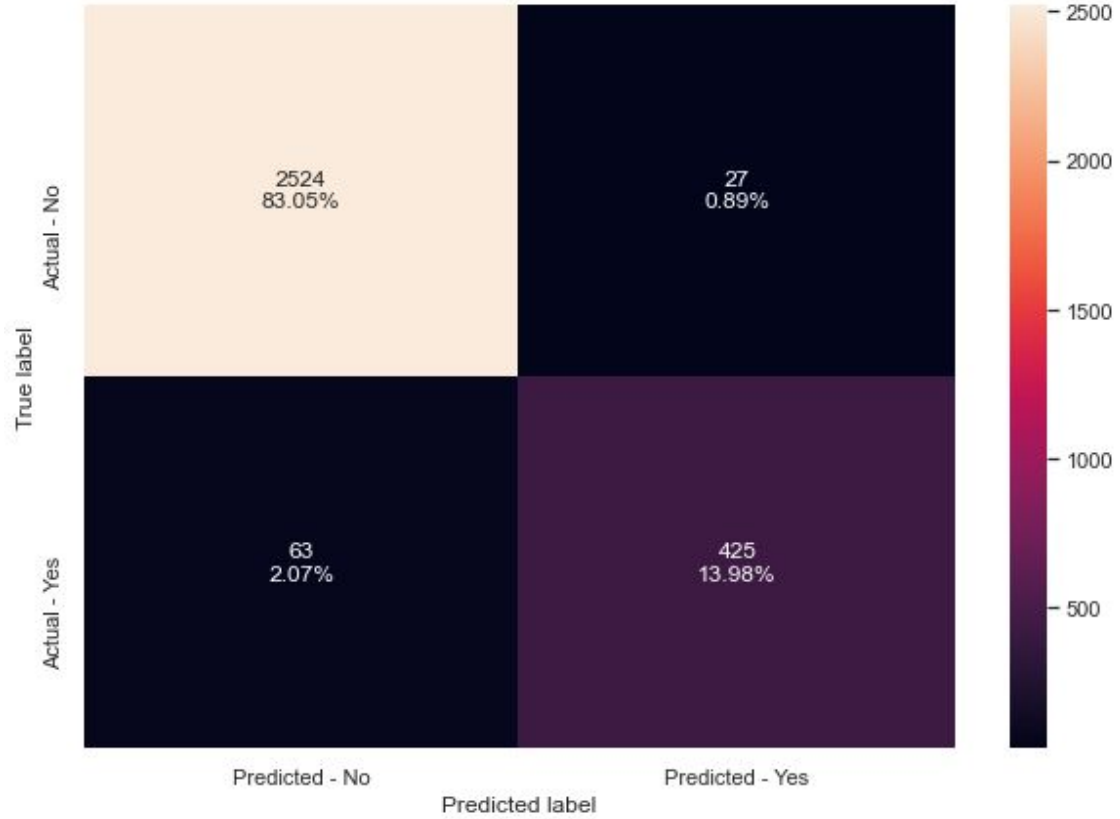
Accuracy on training set : 0.9874435665914221
Accuracy on test set : 0.9703849950641659
Recall on training set : 0.9464442493415277
Recall on test set : 0.8709016393442623
Precision on training set : 0.9746835443037974
Precision on test set : 0.9402654867256637

- The test recall has increased by ~3% as compared to cross-validated recall
- The tuned Gradientboost model is slightly overfitting the training data

# Hyperparameter tuning - GradientBoost with RandomizedSearchCV

Best parameters are
{'gradientboostingclassifier__subsample': 0.8,
'gradientboostingclassifier__n_estimators': 200,
'gradientboostingclassifier__max_features': 0.9}
with CV score=0.8788391684055956

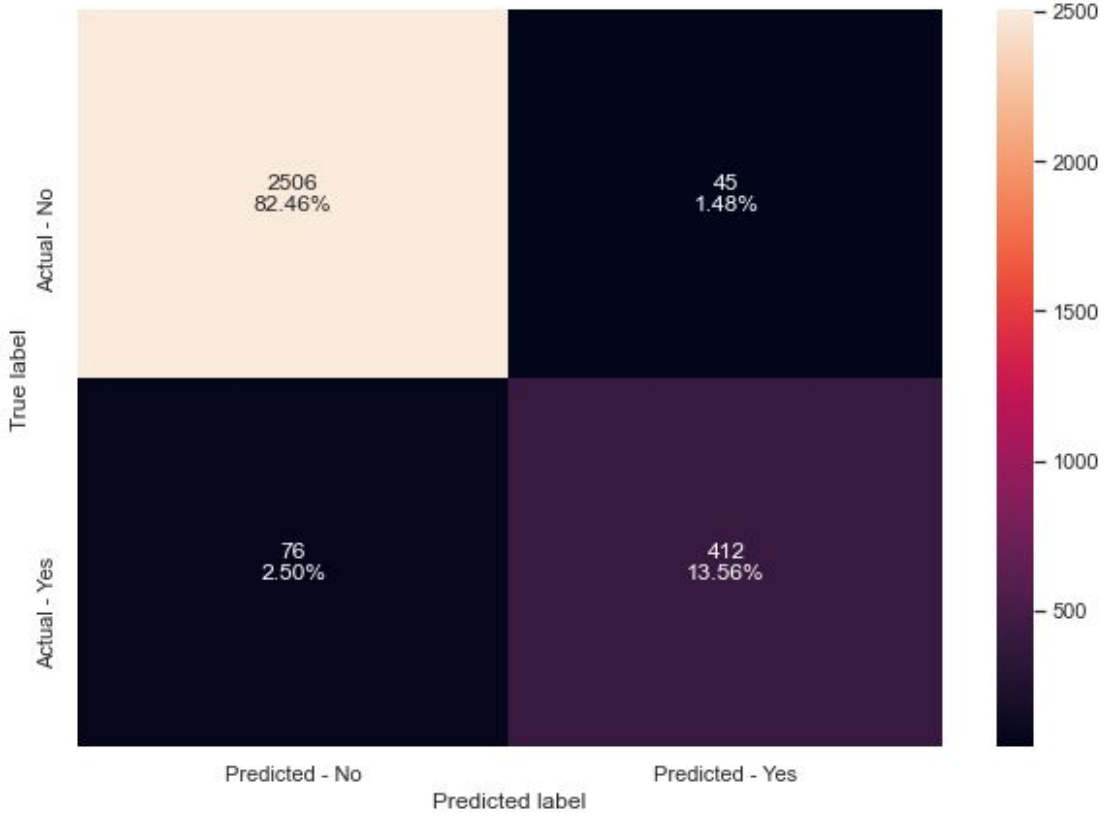
Accuracy on training set : 0.9464442493415277
Accuracy on test set : 0.8709016393442623
Recall on training set : 0.9464442493415277
Recall on test set : 0.8709016393442623
Precision on training set : 0.9746835443037974
Precision on test set : 0.9402654867256637



# Hyperparameter tuning - AdaBoost with GridSearchCV

Accuracy on training set : 0.9922404063205418
Accuracy on test set : 0.9601842711418229
Recall on training set : 0.9701492537313433
Recall on test set : 0.8442622950819673
Precision on training set : 0.9813499111900533
Precision on test set : 0.9015317286652079

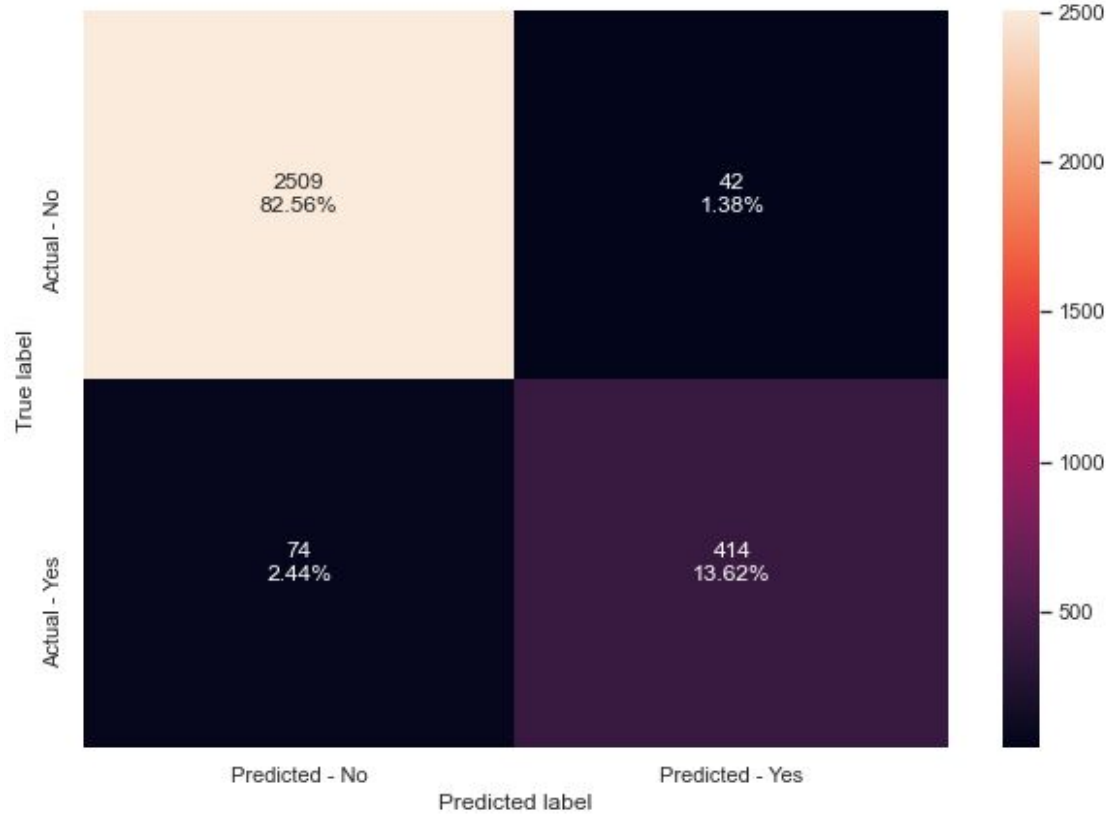
- The test recall has increased by ~3% as compared to cross-validated recall
- The tuned Adaboost model is slightly overfitting the training data



# Hyperparameter tuning - AdaBoost with RandomizedSearchCV

Best parameters are
{'adaboostclassifier__n_estimators': 90,
'adaboostclassifier__learning_rate': 1,
'adaboostclassifier__base_estimator':
DecisionTreeClassifier(max_depth=2,
random_state=1)} with CV
score=0.8849833835690548

Accuracy on training set :
0.9789288849868305
Accuracy on test set : 0.8483606557377049
Recall on training set : 0.9789288849868305
Recall on test set : 0.8483606557377049
Precision on training set :
0.9858532272325375
Precision on test set : 0.9078947368421053



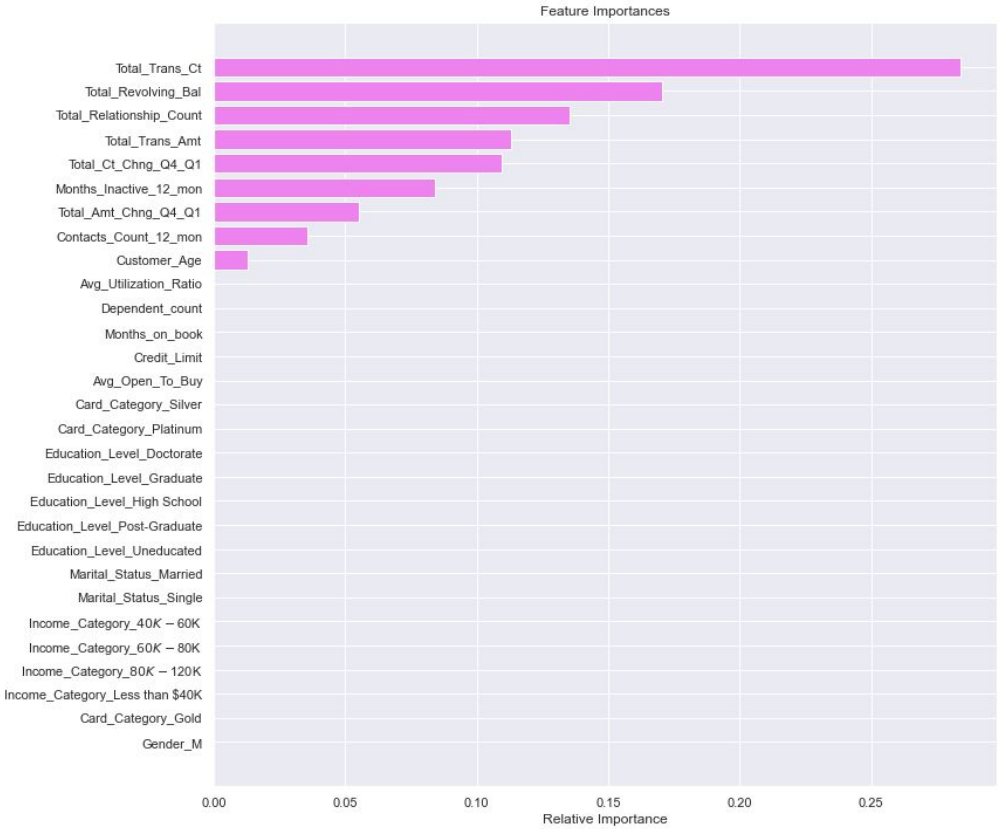
# Model performance evaluation

| Model | | Train_Accuracy | Test_Accuracy | Train_Recall | Test_Recall | Train_Precision | Test_Precision |
|---|---|---|---|---|---|---|---|
| 1 | XGBoost with RandomizedSearchCV | 0.897714 | 0.886805 | 0.985953 | 0.967213 | 0.612991 | 0.590000 |
| 0 | XGBoost with GridSearchCV | 0.967269 | 0.945377 | 0.999122 | 0.956967 | 0.831264 | 0.763072 |
| 2 | GradientBoost with GridSearchCV | 0.987444 | 0.970385 | 0.946444 | 0.870902 | 0.974684 | 0.940265 |
| 3 | GradientBoost with RandomizedSearchCV | 0.946444 | 0.870902 | 0.946444 | 0.870902 | 0.974684 | 0.940265 |
| 5 | Adaboost with RandomizedSearchCV | 0.978929 | 0.848361 | 0.978929 | 0.848361 | 0.985853 | 0.907895 |
| 4 | Adaboost with GridSearchCV | 0.992240 | 0.960184 | 0.970149 | 0.844262 | 0.981350 | 0.901532 |

- The xgboost model tuned using randomised search is giving the best test recall of 0.96 but it has the least train and test precision.
- Compared to Logistic Regression on Undersampled data with 0.811 Test_Recall, xgboost model tuned using randomised search is the best model performer

# Feature importance from the tuned XGBoost Model

Total_Trans_Ct is the most important feature, followed by Total_Revolving_Bal and Total_Relationship_Count of the customer



# Actionable Insights & Recommendations

- Company should target customers with low Total_Trans_Ct -Total Transaction Count (Last 12 months) [Total_Trans_Ct value 50 and lower]
- Company should target customers with low Total_Trans_Amt -Total Transaction Amount (Last 12 months) [Total_Trans_Amt $3000 and lower]
- Company should target customers with low Total_Revolving_Bal: The balance that carries over from one month to the next is the revolving balance [Total_Revolving_Bal value $1400 and lower]
- Company should target customers with low Total_Relationship_Count: Total no. of products held by the customer [Customers with Total_Relationship_Count values 1,2,3 should be targeted.]

greatlearning
*Power Ahead*

# Happy Learning !

Esteban Ordenes

Post Graduate Program in Data Science and Business Analytics

PGP-DSBA-UTA-Dec20-A